(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 869 438 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
07.10.1998 Bulletin 1998/41

(51) Int. Cl.[6]: **G06F 11/14**

(21) Application number: **98105864.7**

(22) Date of filing: **31.03.1998**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(30) Priority: **01.04.1997 JP 98389/97**

(71) Applicant: **Hitachi, Ltd.**
Chiyoda-ku, Tokyo 101-0062 (JP)

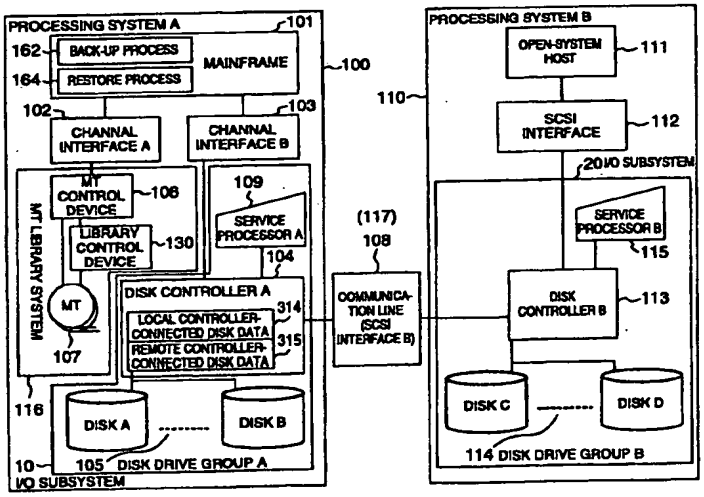
(72) Inventors:
- **Fukuzawa, Yasuko**
 Aoba-ku, Yokohama-shi (JP)
- **Yamamoto, Akira**
 Sagamihara-shi (JP)
- **Nakano, Toshio**
 Chigasaki-shi (JP)

(74) Representative:
Strehl Schübel-Hopf & Partner
Maximilianstrasse 54
80538 München (DE)

(54) Heterogeneous computer system, heterogeneous input/output system and data back-up method for the systems

(57) A heterogeneous computer system, a heterogeneous input/output system and a data back-up method for the systems. An I/O subsystem A (113, 114) for open system and an I/O subsystem B (104, 105) for a mainframe are connected by a communication unit. In order to back up the data from at least a disk connected to the I/O subsystem B in a MT library system and in order to permit the mainframe to access the data in the I/O subsystem B, the I/O subsystem A includes a table (314, 315) for assigning a vacant memory address in a local subsystem to the memory of the I/O subsystem for an open system. A request of variable-length record format received from the mainframe is converted into a fixed-length record format for the subsystem B. The disk designated according to the table is accessed, and the data thus obtained is sent to the mainframe and backed up in the back-up system.

FIG.1



EP 0 869 438 A2

Description

BACKGROUND OF THE INVENTION

The present invention relates to a heterogeneous computer system comprising a host computer and a plurality of I/O subsystems, and more in particular to a method for making it possible to back up the data stored in a memory between a host computer and an I/O subsystem which cannot be directly connected due to the difference in access interface, and a heterogeneous computer system including a plurality of I/O subsystems having different access interfaces connected to the system and the host computer.

In mainframes, a large scale of memory hierarchy (storage hierarchy) including a combination of a plurality of external memories having different processing speeds and different storage capacities is accompanied by a satisfactory data management function and an overall storage management function intended to support an optimum data arrangement and an efficient operation. The IBM's DFSMS (Data Facility Storage Management Subsystem) is an example, which is described in detail in "IBM SYSTEMS JOURNAL, Vol. 28, No. 1, 1989, pp.77-103.

The disk data of the I/O subsystem of the mainframe computer system having the above-mentioned management function can be backed up in a medium such as a magnetic tape or a magnetic tape library capable of storing a large quantity of data with a low cost per bit.

An open system such as a personal computer or a work station, unlike the mainframe, is not equipped with a magnetic tape or a magnetic tape library capable of storing a large quantity of data.

Generally, in an open system such as a personal computer or a work station, a disk is accessed in accordance with a fixed-length record format, while the mainframe accesses a disk in accordance with a variable-length record format called the count key data format.

As a result, the disk subsystem for the mainframe computer is often configured independently of the disk subsystem for the open system.

On the other hand, a technique for transmitting and receiving data between I/O subsystems is disclosed in USP 5155845.

SUMMARY OF THE INVENTION

In a disk subsystem for an open system and a disk subsystem for a mainframe computer which use different host computers, the back-up and other functions are independently operated and managed.

In view of the fact that the open system lacks a medium such as a magnetic tape or a magnetic tape library capable of storing a larger quantity of data, as described above, it is effective to back up the data in the I/O subsystem of the mainframe.

An ordinary disk system for the open system, however, cannot be connected directly to the mainframe due to the difference in the interface thereof.

USP 5155845 fails to disclose how to process the read/write operation for a storage system not directly connected with a host computer.

An object of the present invention is to provide a method and a system for backing up data stored in a memory between a host computer and an I/O subsystem that cannot be connected directly to each other due to the difference in access interface.

Specifically, an object of the invention is to provide a method and a system for backing up data stored in an I/O subsystem of an open system from a mainframe not directly connected with the I/O subsystem.

Another object of the invention is to provide a method and a computer system in which a mainframe is capable of accessing a memory of an I/O subsystem of an open system not directly connected with the mainframe.

Still another object of the invention is to provide a system and a method of access in which two or more I/O subsystems having different interfaces can be connected with a mainframe.

In order to achieve the above-mentioned objects, according to one aspect of the present invention, there is provided a heterogeneous computer system comprising a first host computer, a first I/O subsystem directly connected to the first host computer by an interface of variable-length record format and including at least one external memory, a second host computer, a second I/O subsystem directly connected to the second host computer by an interface of fixed-length record format and including at least one external memory, and a communication unit for connecting the first I/O subsystem and the second I/O subsystem,

wherein the first I/O subsystem includes a table for storing a device address of an external memory, data indicating one of the external memory of the first I/O subsystem and the external memory of the second I/O subsystem to which the device address is assigned, and a device address of the external memory in the second I/O subsystem in the case where the device address is assigned to the external memory of the second I/O subsystem, and

wherein upon receipt of a read/write request conforming with the interface of variable-length record format from

the first host computer and including an address of an external memory to be read from or written into, and upon decision, with reference to the table, that the external memory address included in the read/write request is assigned to the external memory included in the second I/O subsystem, the first I/O subsystem converts the read/write request into a second read/write request conforming with the interface of fixed-length record format and sends the second read/write request to the second I/O subsystem.

According to another aspect of the invention, there is provided a heterogeneous computer system comprising a first host computer, a first I/O subsystem directly connected with the first host computer by an interface of variable-length record format and including at least an external memory, a back-up system connected to the first host computer, a second host computer, a second I/O subsystem directly connected to the second host computer by an interface of fixed-length record format and including at least an external memory, and a communication unit for connecting the first I/O subsystem and the second I/O subsystem,

wherein the first host computer includes means for issuing to the first I/O subsystem a read request conforming with the interface of variable-length record format and containing the address of an external memory from which data is to be read, and backing up the data received from the first I/O subsystem into the back-up system,

wherein the first I/O subsystem includes a table for storing the device address of an external memory, data indicating that one of the external memory of the first and the second I/O subsystems to which the device address is assigned, and the device address of the external memory in the second I/O subsystem in the case where the first device address is assigned to the external memory of the second I/O subsystem, and

wherein upon receipt from the first host computer of a read request conforming with the interface of variable-length record format including an external memory address to be read, and upon decision, with reference to the above-mentioned table, that the device address of the memory address included in the read request is assigned to the external memory included in the second I/O subsystem, the first I/O subsystem converts the read request into a second read request conforming with the fixed-length interface and sends the second read request to the second I/O subsystem while at the same time sending to the first host computer the data received from the second I/O subsystem.

According to still another aspect of the invention, there is provided a heterogeneous computer system comprising a first host computer, a first I/O subsystem directly connected to the first host computer by an interface of variable-length record format and including at least one external memory, a back-up system connected to the first host computer, a second host computer, a second I/O subsystem directly connected to the second host computer by an interface of fixed-length record format and including at least one external memory, and a communication unit for connecting the first I/O subsystem and the second I/O subsystem,

wherein the first host computer includes means for issuing to the first I/O subsystem a write request conforming with the interface of variable-length record format including the address of an external memory into which data is to be written, and sending the data read from the back-up system to the first I/O subsystem,

wherein the first I/O subsystem includes a table containing a device address of the external memory, data indicating one of the external memories of the first and the second I/O subsystems to which the device address is assigned, and a device address of the external memory in the second I/O subsystem in the case where the first device address is assigned to the external memory of the second I/O subsystem, and

wherein upon receipt from the first host computer of a write request conforming with the interface of variable-length record format including the device address of an external memory to be written into, and upon decision, with reference to the table, that the address of the external memory included in the write request is assigned to the external memory included in the second I/O subsystem, the first I/O subsystem converts the write request into a second write request conforming with the interface of fixed-length record format, sends the second write request to the second I/O subsystem while at the same time sending the data received from the first host computer to the second I/O subsystem.

According to yet another aspect of the invention, there is provided a heterogeneous I/O system for use with a host computer connected thereto, comprising a first I/O subsystem including at least an external memory, and a second I/O subsystem connected to the first I/O subsystem and including at least an external memory,

wherein the first I/O subsystem includes a table for storing a device address of an external memory, data indicating one of the external memories of the first and the second I/O subsystems to which the device address is assigned, and a device address of the external memory in the second I/O subsystem in the case where the first device address is assigned to the external memory of the second I/O subsystem,

wherein upon receipt of a read/write request designating the device address of an external memory to be read from or written into by the host computer, and upon decision, with reference to the table, that the external memory address in the designated address is assigned to the external memory included in the second I/O subsystem, the first I/O subsystem sends the read/write request to the second I/O subsystem.

According to a further aspect of the invention, there is provided a heterogeneous I/O system for use with a host computer connected thereto, comprising a first I/O subsystem having an interface of variable-length record format and including at least one external memory, a second I/O subsystem having an interface of fixed-length record format and including at least one external memory, and a communication unit for connecting the first I/O subsystem and the second

I/O subsystem,

wherein the first I/O subsystem includes a table for storing a device address of an external memory, data indicating one of the external memories of the first and the second I/O subsystems to which the device address is assigned, and a device address of the external memory in the second I/O subsystem in the case where the first device address is assigned to the external memory of the second I/O subsystem; and

wherein upon receipt from the host computer of a read/write request conforming with the interface of variable-length record format including the address of an external memory to be read from or written into, and upon decision, with reference to the table, that the external memory address included in the read/write request is assigned to the external memory included in the second I/O subsystem, the first I/O subsystem converts the read/write request into a second read/write request conforming with the interface of fixed-length record format and sends it to the second I/O subsystem.

According to an embodiment of the invention, there is provided a heterogeneous computer system, wherein an I/O subsystem for an open system is connected with an I/O subsystem for a mainframe by a communication unit, wherein in order to access data in the I/O subsystem for an open system from the mainframe for enabling the data in the disk connected to the I/O subsystem for the open system to be backed up in a MT library system, a table is prepared for assigning a vacant address of the memory in the local subsystem to the memory of the I/O subsystem for the open system, wherein a request of variable-length record format received from the mainframe is converted into a request of fixed-length record format for the open system, wherein the disk designated according to the table is accessed, and wherein the data thus obtained is sent to the mainframe and backed up in the back-up system.

This configuration can back up the data of an I/O subsystem for an open system in a back-up system under the management of a mainframe not directly connected to the particular I/O subsystem.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing a configuration of a heterogeneous computer system according to an embodiment of the present invention.

Fig. 2 is a block diagram showing a configuration of a heterogeneous computer system according to another embodiment of the invention.

Fig. 3 is a block diagram showing a configuration of a disk controller of the heterogeneous computer system shown in Figs. 1 and 2.

Fig. 4 is a diagram showing a configuration of a local controller-connected disk data (table) for the systems shown in Figs. 1 and 2.

Fig. 5 is a diagram showing a configuration of a remote controller-connected disk data (table) for the systems shown in Figs. 1 and 2.

Fig. 6 is a diagram showing the interconnection of disk devices as viewed from the mainframe.

Fig. 7 is a diagram showing an example of the processing flow of a disk controller A in the case where the data in an I/O subsystem for an open system is backed up in a MT library system of the mainframe.

Fig. 8 is a diagram showing an example of the processing flow of a disk controller A in the case where data are restored in an I/O subsystem for an open system from a MT library system of the mainframe.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiments of the invention will be described below with reference to the accompanying drawings.

Fig. 1 is a diagram showing a configuration of a computer system according to an embodiment of the invention.

A processing system A 100 includes a mainframe 101, a channel interface A 102, a channel interface B 103, a magnetic tape (MT) controller 106, a magnetic tape library controller 130, a magnetic tape library 107, a disk controller A 104, a disk drive group A 105 and a service processor 109. A back-up processing device 162 and a restore processing device 164 are mounted on the mainframe 101.

The mainframe 101 accesses the disk controller A 104 through the channel interface B 103 conforming with a variable-length record format called the count-key-data format.

The count-key-data format is a record format in which a record constituting a unit of read/write operation is configured of three fields including a count field, a key field and a data field.

A record ID is stored in the count field, a key data for accessing the record is stored in the key field, and the data used by an application program is stored in the data field.

In the description that follows, the magnetic tape (MT) controller 106, the magnetic tape library controller 130 and the magnetic tape library 107 are collectively referred to as a MT library system 116. The disk controller A 104 and the disk drive group A 105 constitute an I/O subsystem 10 connected to the mainframe 101. In similar fashion, the disk controller B 113 and the disk drive group B 114 constitute an I/O subsystem 20 connected to a host 111 for an open system.

An optical disk or the like, as well as a magnetic disk, constitutes a rank of storage hierarchy connected through the

channel interface. The following description refers to the case in which the MT library system 116 is connected.

The disk controller A 104 contains local controller-connected disk data 314 and remote controller-connected disk data 315.

The local controller-connected disk data 314 and the remote controller-connected disk data 315 are data provided for making it possible for the mainframe to access a disk device of the I/O subsystem not directly connected. Specifically, the data 314, 415 are a table for assigning a vacant address of the memory in the local I/O subsystem 10 for the processing system A to the memory of the I/O subsystem for the open system so that the data in the I/O subsystem 20 for the processing system B can be accessed from the mainframe 101. The data 314, 315 will be described in detail later.

The processing system B 110 includes a host 111 for the open system, a SCSI (small computer system interface) 112, the disk controller B 113, the disk drive group B 114 and a service processor B 115.

The host 111 for the open system accesses the disk controller B 113 through the SCSI 112 having a fixed-length record which is a unit of read/write operation.

The disk controller A 104 and the disk controller B 113 are connected by a communication line 108. The communication line 108 can be, for example, a SCSI cable B 117.

In the description that follows, the count-key-data format will be called the CKD format, and the fixed-length block format a FBA (fixed block architecture) format.

Also, the record of the CKD format will be referred to as the CKD record, and the record of the FBA format as the FBA record.

Fig. 2 is a diagram showing another example of a computer system according to the invention, in which a single I/O subsystem for the mainframe is connected with two or more I/O subsystems for an open system.

In a processing system X 120, the interfaces of an open system host X 121 and a disk controller X 123 are connected to each other by a fiber channel interface 122. The fiber channel interface 122 is an optical fiber cable which can increase the length of connection between a host and a control device.

In many case, however, a fiber channel interface based on SCSI is employed between a host and a control device.

Also, an interface like a fiber channel interface X 126 can be used to connect a disk controller X 123 and the disk controller B 113.

The data back-up system in the configuration of Fig. 2 is an expansion of the data back-up system in the configuration of Fig. 1.

The fundamental operation of each system is such that the mainframe 101 and the hosts 111, 121 for the open system access the magnetic tape library 107 constituting an external memory or the disk drive group A 105, the disk drive group B 114 and the disk drive group X 124 through each interface.

The process in the mainframe 101 establishes a route to the data stored externally through each interface under the control of an arbitrary operating system such as Hitachi's VOS3 (virtual-storage operating system 3) for supporting the channel interface, while the process in the host for the open system establishes a route to the externally-stored data through each interface under the control of an arbitrary operating system such as UNIX (a registered trade mark owned by X/Open in U.S.A. and other countries) for supporting the SCSI.

Fig. 3 is a diagram showing a configuration of the disk controller A 104.

The disk controller A 104 includes a MPU 302 for executing a control system process 307 of the disk controller, a memory 301, a host data transfer device 303, a disk/cache device 304, an inter-I/O subsystem data transfer device 305, a data transfer device 306 and a bus 308 for connecting these devices.

The control system process 307 operates in a multitask or multiprocessor environment.

The memory 301 includes various microprograms 312 and various data 313.

Especially, the disk controller A 104 has stored therein the local controller-connected data 314 and the remote controller-connected disk data 315, as described above with reference to Fig. 1.

The disk controller B 113 and the disk controller X 123 have a configuration similar to the disk controller A 104 and will not be described in detail.

The disk controller B 113 and the disk controller X 123, however, are not required to contain the local controller-connected disk data 314 and the remote controller-connected disk data 315.

The local controller-connected disk data 314 is the data indicating the connections of the controllers and the like, and stored in the memory 301 of the disk controller A 104. The local controller-connected disk data 314 exists as the data corresponding to each disk device.

The local controller-connected disk data 314 is shown in Fig. 4.

The device address is an identifier (ID) for discriminating a disk device to be read from or written into by a host computer such as the mainframe 101, and is the data also contained in the read/write request issued by the host computer such as the mainframe 101.

Local controller connection data 401 is the data indicating whether or not the disk drive corresponding to the controller-connected disk data 314 is actually connected to a controller.

A remote controller connection pointer 402 indicates whether or not the controller-connected disk data 314 is assigned to a disk drive connected to a remote controller.

In the case where the such data is assigned to a disk drive connected to a remote controller, the pointer indicates a corresponding remote controller-connected disk data 315. Otherwise, the pointer assumes a null value.

In the case where the remote controller connection pointer 402 is valid (i.e. in the case where the particular device address 400 is assigned to a disk device connected to a remote controller), it represents the state in which the local controller connection data 401 is not assigned.

In the case where the remote controller connection pointer 402 is invalid (i.e. in the case where the device address 400 is not assigned to a disk drive connected to a remote controller), on the other hand, the local controller connection data 401 may indicate the state of no-assignment.

In other words, the device address 400 may be assigned to neither a disk device connected to a local controller nor a disk device connected to a remote controller.

An attribute 403 is the data unique to a device including the interface, the function, the data format and the block length of the disk drive.

The local controller-connected disk data 315 shown in Fig. 5 is the data corresponding to a disk drive not directly connected to the disk controller A 104.

It follows therefore that the remote controller-connected disk data 315, on the other hand, is pointed to by any one of the local controller-connected disk data 314.

A connection controller address 500 represents the address of a controller connected with a disk device corresponding to the remote controller-connected disk data 315. According to this embodiment, the address of the disk controller B 113 is stored as the connection controller address 500.

A disk address 501 represents the address assigned in the controller actually connected to a corresponding disk drive.

The local controller-connected disk data 314 and the remote controller-connected disk data 315 are set from the service processor 109.

According to this embodiment, the mainframe 101 recognizes that the disk drive group B 114 (disks C and D) is also connected to the disk controller A 104 through the disk controller B 113, as shown in Fig. 6, taking advantage of the local controller-connected disk data 314 and the remote controller-connected disk data 315 shown in Figs. 4 and 5.

This is because of the fact that the vacant address of disk drive available in the disk controller A 104 is assigned by the disk controller A 104 to a disk drive of the I/O subsystem for an open system.

Now, the back-up processing will be described with reference to Figs. 1, 7 and 8.

Specifically, in Fig. 1 the back-up process 162 on the mainframe 101 causes the data in the disk device group B 114 of the open system of the processing system B to be backed up in the MT library system 116 through the disk controller A 104 and the mainframe 101 of the processing system A.

Conversely, the data backed up in the MT library system 116 is restored in the disk drive group B 114 of the open system of the processing system B through the mainframe 101 and the disk controller A 104 of the processing system A.

The back-up operation and the restoration described above are executed in response to a command from the mainframe 101.

First, an explanation will be given of the case in which the data in the disk drive group B 114 of the open system for the processing system B is backed up in the MT library system 116 through the disk controller A 104 and the mainframe 101 of the processing system A.

As already described above, the mainframe 101 has recognized that the disk drive group B 114 (disks C and D) are also connected to the disk drive A 104.

Therefore, the operation of the mainframe 101, which is simply to issue a read request to the disk controller A 104 and back up the received data in the MT library system 116, will not be described specifically.

In the case of backing up data into the MT library system 116, the mainframe 101 issues a read request to the disk controller A 104. The disk controller A 104 executes the process in accordance with the flowchart of Fig. 7 in response to a read request from the mainframe 101.

First, step 700 finds out a corresponding local controller-connected disk data 314 from the address of the disk drive designated in the read request.

Step 701 checks whether the designated disk drive is connected to the disk controller A 104 or not.

In the case where the disk drive is connected to the disk controller A 104, step 702 reads the corresponding data from the particular disk drive.

In the case where the disk drive is not connected to the disk controller A104, in contrast, step 703 checks whether the designated disk drive is connected to a remote disk controller (disk controller B 113). In other words, it checks whether the remote controller connection pointer 402 assumes a null value.

In the case where the check result shows that the remote controller connection pointer 402 assumes a null value

indicating that the designated disk drive is not connected to the remote disk controller, an error is reported in step 704.

The operation specifically related to the invention is represented by step 705 and subsequent steps executed in the case where a designated disk drive is connected to a remote disk controller (disk controller B 113).

First, in the case where the check result shows that the remote controller connection pointer 402 does not assume the null value indicating that the designated disk drive is connected to a remote disk controller, step 705 finds out the remote controller-connected disk data 315 corresponding to the designated disk drive based on the remote controller connection pointer 402. Then, the address of the disk controller (disk controller B 113) actually connected with the designated disk drive and the address of the disk drive in the disk drive group B connected to the particular disk controller B 113 are acquired on the basis of the remote controller-connected disk data 315 found as above.

Then, step 706 converts the address of the data to be read which has been received in the read request into the format of the disk drive connected to the disk controller B 113.

In a read/write request from the mainframe 101, the address of data to be read or written is normally designated by the cylinder number, the head number and the record number according to the CKD format.

The record address expressed by the cylinder number, the head number and the record number will hereinafter be called CCHHR.

The disk drive connected to the disk controller B 113, on the other hand, has an access interface designated by LBA (logical block address) in accordance with the FBA format.

Consequently, step 706 converts the access address of the data to be read from CKD format to FBA format.

The conversion formula is given, for example, by

$$LBA = (CC \times \text{number of heads} + HH) \times \text{track length} + \text{record number} \times \text{record length}$$

According to this embodiment, the disk controller A 104 and the disk controller B 113 may have the same interface, in which case the conversion of the input/output interface format is not required.

Step 707 issues a request to the disk controller B 113 to read the data from the area of the corresponding disk drive calculated in step 706.

Step 708 waits for the arrival of the requested data from the disk controller B 113.

Step 709 sends the data received from the disk controller B 113 to the main frame 101 thereby to complete the process.

The disk controller B 113 simply reads the data requested by the disk controller A 104 from a disk drive, and sends it to the disk controller A 104. This process, therefore, is not described specifically in the processing flow.

Next, an explanation will be given of a case in which data backed up in the MT library system 116 is restored by the restore process 164 on the mainframe 101 in the disk drive group B 114 of the open system of the processing system B through the disk controller A 104 and the mainframe 101 of the processing system A.

As described already above, the mainframe 101 has recognized that the disk drive group B 113 (disks C and D) are also connected to the disk controller A 104.

Therefore, no explanation will be given of the operation of the mainframe 101 which is simply to issue a write request to the disk controller A 104 to write the data read from the MT library system 116. Upon receipt of a write request from the mainframe 101, the disk controller A 104 executes the process in accordance with the flowchart of Fig. 8.

In the processing flow of Fig. 8, steps 800 to 801, 803 to 806 are similar to steps 700 to 701, 703 to 706 in Fig. 7, respectively, and therefore will not be explained. Also, step 802 is normally the write operation, since the request from the mainframe 101 is a write request. Only the parts different from Fig. 7 will be described below.

Step 807 issues a request to the disk controller B 113 to write data in the area of the corresponding disk drive calculated in step 807.

Next, in step 808, the write data is received from the mainframe 101 and sent to the disk controller B 113.

Then, step 809 waits for a report on the completion of the write request from the disk controller B 113, and upon receipt of the completion report, sends it to the mainframe 101 thereby to complete the process.

The disk controller B 113 simply reads the data requested by the disk controller A 104 from the corresponding disk drive and sends it to the disk controller A 104. The related processing flow, therefore, is not shown specifically.

The foregoing description concerns a system for backing up data of the disk drive group B 114 of the open system of the processing system B by the processing system A. As another embodiment, a heterogeneous I/O subsystem can be configured in which only the disk controller B and the disk drive group B are connected to the processing system A and the mainframe is connected with two I/O subsystems having different interfaces. In such a case, three or more instead of two I/O subsystems can be connected.

The above-mentioned embodiment permits data to be backed up between I/O subsystems having different access interfaces.

As a result, data stored in an I/O subsystem for an open system can be backed up into an I/O subsystem for the

mainframe.

Also, the back-up mechanism of the mainframe includes a large-capacity, high-performance and high-reliability MT library system. The data of the I/O subsystem for an open system, therefore, can be backed up by a mainframe back-up mechanism high in performance and reliability.

Further, different I/O subsystems can be connected to the mainframe.

Claims

1. A heterogeneous computer system comprising:

a first host computer (101);
a first I/O subsystem (104, 105) connected by an interface of variable-length record format (103) to said first host computer and including at least an external memory;
a second host computer (111);
a second I/O subsystem (113, 114) connected to said second host computer by an interface of fixed-length record format (112); and
a communication unit (108) for connecting said first I/O subsystem and said second I/O subsystem;
wherein said first I/O subsystem includes

a table (314, 315) for storing a device address (400) of an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a unit address (501) of said external memory in said second I/O subsystem when said unit address is assigned to the external memory of said second I/O subsystem, and
means (707, 807) for receiving from said first host computer a read/write request conforming with said interface of variable-length record format including the address of the external memory to be read from or written into, and upon decision, with reference to said table, that said address of said external memory included in said first read/write request is assigned to the external memory included in said second I/O subsystem, converting said read/write request into a read/write request conforming with said interface of fixed-length record format and sending said second read/write request to said second I/O subsystem.

2. A heterogeneous computer system comprising:

a first host computer (101);
a first I/O subsystem (104, 105) connected by an interface of variable-length record format (103) to said first host computer and including at least one external memory;
a second host computer (111);
a second I/O subsystem (113, 114) connected to said second host computer by an interface of fixed-length record format (112); and
a communication mechanism (108) for connecting said first I/O subsystem and said second I/O subsystem;
wherein said first host computer includes means for issuing to said first I/O subsystem a read request conforming with said interface of variable-length record format including the address of the external memory from which data is to be read, and backing up the data received from said first I/O subsystem in said back-up system;
wherein said first I/O subsystem includes

a table (314, 315) for storing a device address (400) of an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said unit address is assigned to the external memory of said second I/O subsystem,
means (707, 807) for receiving from said first host computer a read request conforming with said interface of variable-length record format including the address of the external memory to be read from, and upon decision, with reference to said table, that said address of said external memory included in said first read request is assigned to the external memory included in said second I/O subsystem with reference to said table, converting said read request into a second read request conforming with said interface of fixed-length record format, sending said second read request to said second I/O subsystem, and sending the data received from said second I/O subsystem to said first host computer,
means for backing up said data received by said first host computer in said back-up system.

3. A heterogeneous computer system comprising:

a first host computer (101);
a first I/O subsystem (104, 105) connected by an interface of variable-length record format (103) to said first host computer and including at least an external memory;
a second host computer (111);
a second I/O subsystem (113, 114) connected to said second host computer by an interface of fixed-length record format (112); and
a communication unit (108) for connecting said first I/O subsystem and said second I/O subsystem;
wherein said first host computer includes means for issuing to said first I/O subsystem a write request conforming with said interface of variable-length record format including the address of the external memory to be written into, and sending the data read from said back-up system to said first I/O subsystem;
wherein said first I/O subsystem includes

a table (314, 315) for storing a device address (400) of an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said device address is assigned to the external memory of said second I/O subsystem,
means (707, 807) for receiving from said first host computer a write request conforming with said interface of variable-length record format including the address of the external memory to be written into, and upon decision, with reference to said table, that said address of said external memory included in said first write request is assigned to the external memory included in said second I/O subsystem with reference to said table, converting said write request into a second write request conforming with said interface of fixed-length record format, sending said second write request to said second I/O subsystem, and sending the data received from said first host computer to said second I/O subsystem.

4. A heterogeneous input/output system for use with a host computer connected thereto, comprising:

a first I/O subsystem including at least an external memory, and a second I/O subsystem connected to said first I/O subsystem and including at least an external memory;
wherein said first I/O subsystem includes

a table (314, 315) for storing a device address (400) for an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said device address is assigned to the external memory of said second I/O subsystem, and
means (706, 806) for receiving a read/write request from said host computer designating an address of the external memory to be read from or written into, and sending said read/write request to said second I/O subsystem, upon decision, with reference to said table, that said address of said external memory designated in said read/write request is assigned to the external memory included in said second I/O subsystem.

5. A heterogeneous input/output system for use with a host computer connected thereto, comprising:

a first I/O subsystem (104, 105) having an interface of variable-length record format and including at least an external memory, and a second I/O subsystem (113, 114) having an interface of fixed-length record format and including at least an external memory, and a communication unit (108) for connecting said first I/O subsystem and said second I/O subsystem;
wherein said first I/O subsystem includes

a table (314, 315) for storing a device address (400) of an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said device address is assigned to the external memory of said second I/O subsystem, and
means (706, 806) for receiving a read/write request from said host computer designating an address of the external memory to be read from or written into, and upon decision, with reference to said table, that said address of said external memory designated in said read/write request is assigned to the external memory included in said second I/O subsystem, converting said read/write request into a read/write request con-

forming with said fixed-length record format and sending said read/write request to said second I/O subsystem.

6. A method of accessing an external memory for a heterogeneous computer system comprising a first host computer (101), a first I/O subsystem (104, 105) connected by an interface of variable-length record format (103) to said first host computer and including at least an external memory, a second host computer (111), a second I/O subsystem (113, 114) connected to said second host computer by an interface of fixed-length record format (112) and including at least an external memory, and a communication unit (108) for connecting said first I/O subsystem and said second I/O subsystem, said method comprising the steps of:

storing in the internal table of said first I/O subsystem a device address (400) of an external memory, data (401) indicating the external memory of that one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said device address is assigned to the external memory of said second I/O subsystem; and

said first I/O subsystem receiving from said first host computer a read/write request conforming with said interface of variable-length record format including the address of said external memory to be read from or written into and deciding, with reference to said table, whether the address of the external memory included in said read/write request is assigned to the external memory included in said second I/O subsystem; and converting said read/write request into a read/write request conforming with the interface of fixed-length record format and sending said second read/write request to said second I/O subsystem in the case where said address of said external memory is assigned to said external memory included in said second I/O subsystem.

7. A method of backing up the data in a heterogeneous computer system comprising a first host computer (101), a first I/O subsystem (104, 105) connected by an interface of variable-length record format (103) to said first host computer and including at least an external memory, a back-up system connected to said first host computer, a second host computer (111), a second I/O subsystem (113, 114) connected to said second host computer by an interface of fixed-length record format (112) and including at least one external memory, and a communication unit (108) for connecting said first I/O subsystem and said second I/O subsystem, said first host computer including means for issuing a read request to said first I/O subsystem and backing up the data received from said first I/O subsystem in said back-up system, said method comprising the steps of:

storing in the internal table of said first I/O subsystem a device address (400) of an external memory, data (401) indicating the external memory of that one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said first device address is assigned to the external memory of said second I/O subsystem; and

said first I/O subsystem receiving from said first host computer a read request conforming with said interface of variable-length record format including the address of said external memory to be read from and deciding, with reference to said table, whether the address of the external memory included in said read request is assigned to the external memory included in said second I/O subsystem;
converting said read request into a read request conforming with the interface of fixed-length record format and sending said second read request to said second I/O subsystem in the case where said address of said external memory is assigned to said external memory included in said second I/O subsystem;

said first I/O subsystem receiving the read data from said second I/O subsystem, and sending said received read data to said first host computer; and
backing up the data received by said first host computer in said back-up system.

8. A method of restoring the back-up data in a heterogeneous computer system comprising a first host computer (101), a first I/O subsystem (104, 105) connected by an interface of variable-length record format (103) to said first host computer and including at least an external memory, a back-up system connected to said first host computer, a second host computer (111), a second I/O subsystem (113, 114) connected to said second host computer by an interface of fixed-length record format (112) and including at least one external memory, and a communication unit (108) for connecting said first I/O subsystem and said second I/O subsystem, said first host computer including means for issuing a write request to said first I/O subsystem and sending the data read from said back-up system to said first I/O subsystem, said method comprising the steps of:

storing in the internal table of said first I/O subsystem a device address (400) of an external memory, data (401)

indicating the external memory of that one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said device address is assigned to the external memory of said second I/O subsystem; and

said second I/O subsystem receiving from said first host computer a write request conforming with said interface of variable-length record format including the address of an external memory to be written into and deciding, with reference to said table, whether the external memory address included in said write request is assigned to the external memory included in said second I/O subsystem;

converting said write request into a write request conforming with the interface of fixed-length record format and sending said second write request to said second I/O subsystem in the case where said external memory address is assigned to said external memory included in said second I/O subsystem; and

said first I/O subsystem sending the data read from said back-up system and received from said first host computer to said second I/O subsystem.

9. A method of accessing an external memory in a heterogeneous input/output system for use with a host computer connected thereto, comprising a first I/O subsystem including at least an external memory, and a second I/O subsystem connected to said first I/O subsystem and including at least an external memory, said method comprising the steps of:

storing in the internal table (314, 315) of said first I/O subsystem a device address (400) of an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said first device address is assigned to the external memory of said second I/O subsystem,

said first I/O subsystem receiving from said host computer a read/write request designating the address of the external memory to be read from or written into and deciding, with reference to said table, whether said designated external memory address is assigned to the external memory included in said second I/O subsystem, and

sending said read/write request to said second I/O subsystem in the case where said designated external memory address is assigned to the external memory included in said second I/O subsystem.

10. A method of accessing an external memory in a heterogeneous input/output system for use with a host computer connected thereto, comprising a first I/O subsystem having an interface of variable-length record format and including at least an external memory, a second I/O subsystem having an interface of fixed-length record format and including at least an external memory, and a communication unit for connecting said first I/O subsystem and said second I/O subsystem, said method comprising the steps of:

storing in the internal table (314, 315) of said first I/O subsystem a device address (400) of an external memory, data (401) indicating one of said first I/O subsystem and said second I/O subsystem to which said device address is assigned, and a device address (501) of said external memory in said second I/O subsystem in the case where said first device address is assigned to the external memory of said second I/O subsystem,

said first I/O subsystem receiving from said host computer a read/write request conforming with said interface of variable-length record format including the address of an external memory to be read from or written into, and deciding, with reference to said table, whether the external memory address included in said read/write request is assigned to the external memory included in said second I/O subsystem, and

converting said read/write request into a second read/write request conforming with said interface of fixed-length record format and sending said second read/write request to said second I/O subsystem in the case where said external memory address is assigned to the external memory included in said second I/O subsystem.

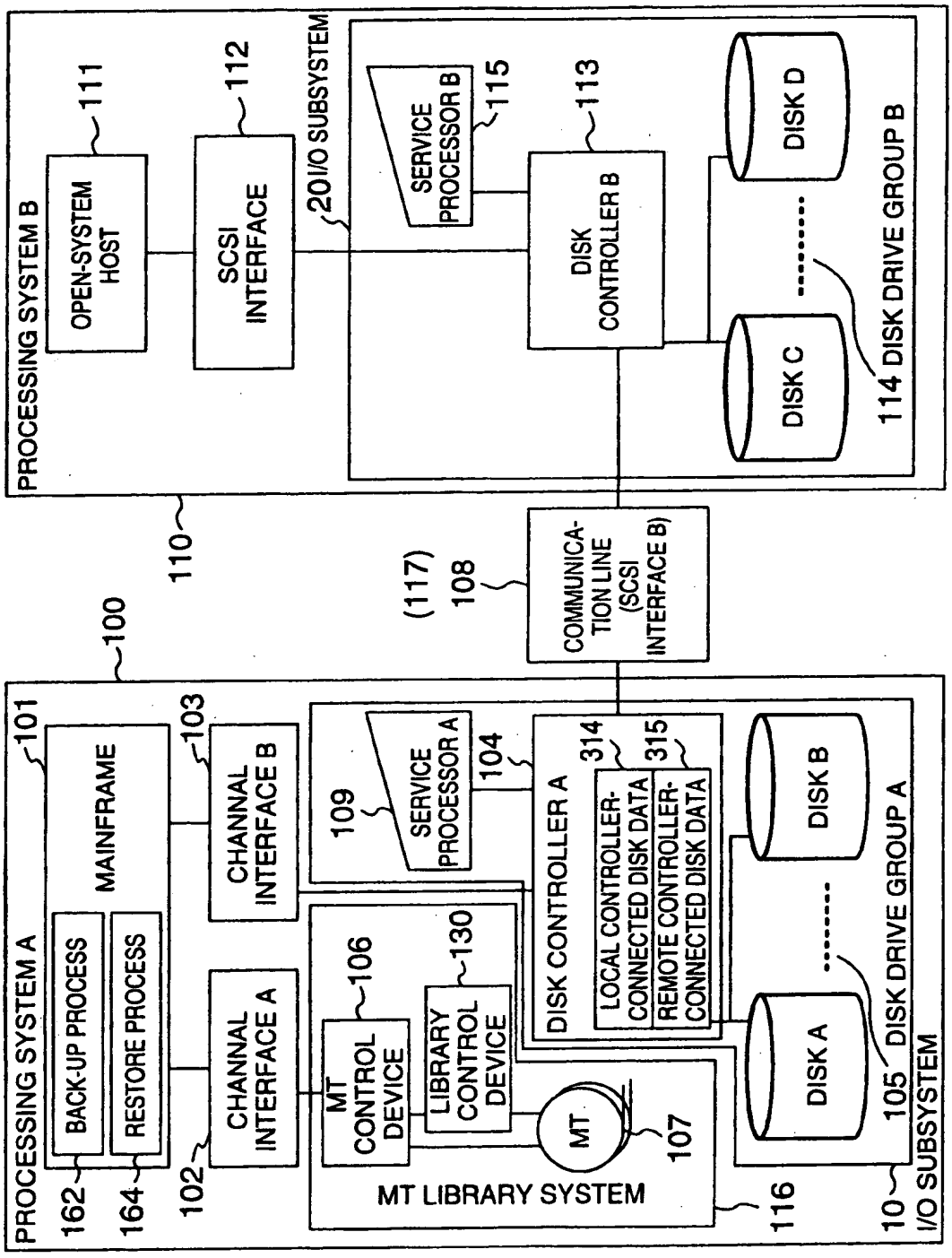

PROCESSING SYSTEM B
OPEN-SYSTEM HOST
111
SCSI INTERFACE
112
20 I/O SUBSYSTEM
SERVICE PROCESSOR B
115
DISK CONTROLLER B
113
DISK D
DISK C
114 DISK DRIVE GROUP B
110
(117)
108
COMMUNICATION LINE (SCSI INTERFACE B)
100
PROCESSING SYSTEM A
101
MAINFRAME
BACK-UP PROCESS
RESTORE PROCESS
162
164
102
CHANNAL INTERFACE A
103
CHANNAL INTERFACE B
109
SERVICE PROCESSOR A
104
DISK CONTROLLER A
314
315
LOCAL CONTROLLER-CONNECTED DISK DATA
REMOTE CONTROLLER-CONNECTED DISK DATA
106
MT CONTROL DEVICE
130
LIBRARY CONTROL DEVICE
MT
107
MT LIBRARY SYSTEM
116
DISK B
DISK A
105
DISK DRIVE GROUP A
10
I/O SUBSYSTEM

FIG.1

FIG.2

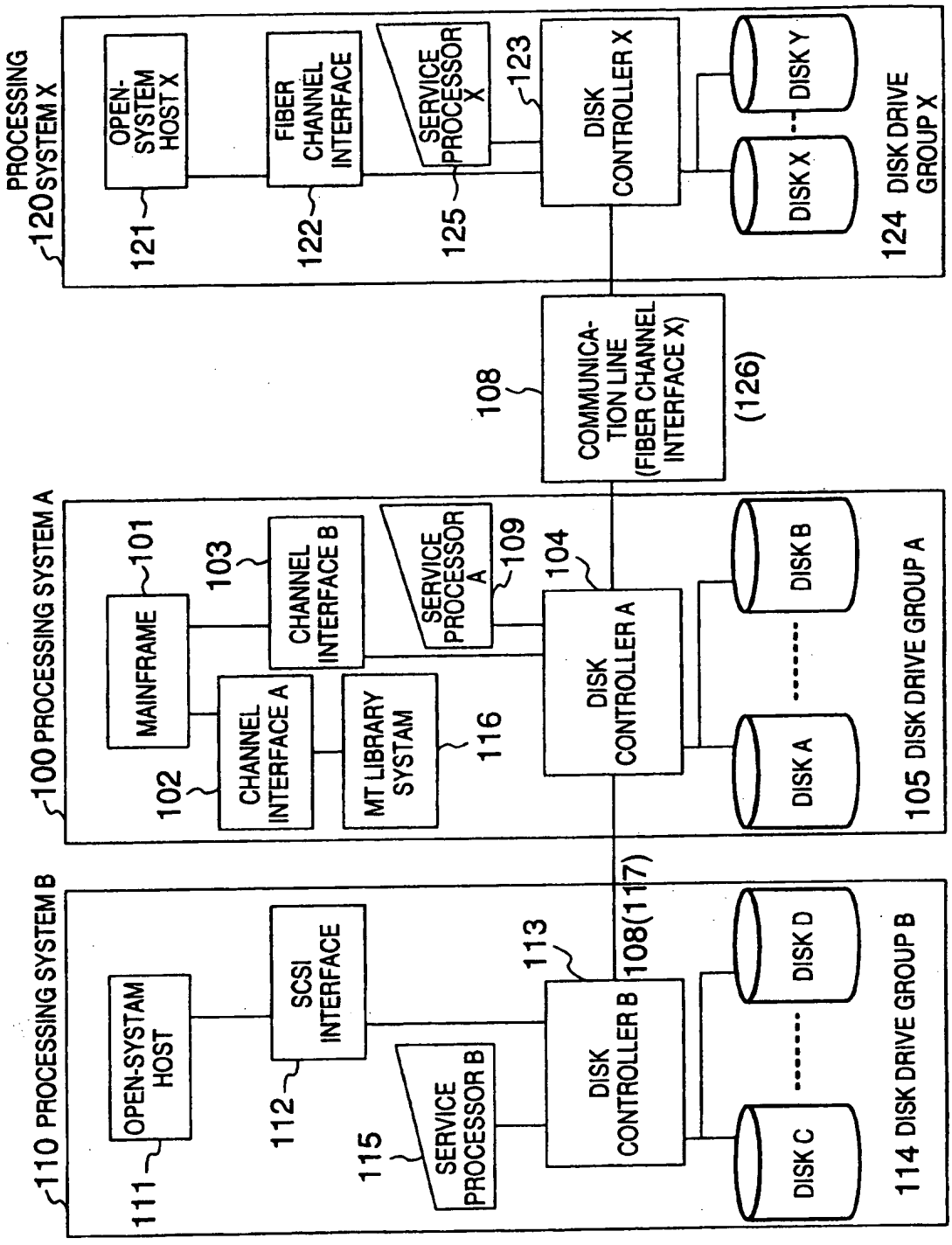

PROCESSING SYSTEM X
120
OPEN-SYSTEM HOST X
121
FIBER CHANNEL INTERFACE
122
SERVICE PROCESSOR X
123
125
DISK CONTROLLER X
DISK Y
DISK X
124 DISK DRIVE GROUP X
108
COMMUNICATION LINE (FIBER CHANNEL INTERFACE X)
(126)
100 PROCESSING SYSTEM A
MAINFRAME
101
102
CHANNEL INTERFACE A
103
CHANNEL INTERFACE B
MT LIBRARY SYSTAM
116
SERVICE PROCESSOR A
109
104
DISK CONTROLLER A
DISK B
DISK A
105 DISK DRIVE GROUP A
110 PROCESSING SYSTEM B
OPEN-SYSTAM HOST
111
SCSI INTERFACE
112
SERVICE PROCESSOR B
115
113
DISK CONTROLLER B
108(117)
DISK D
DISK C
114 DISK DRIVE GROUP B

FIG.3

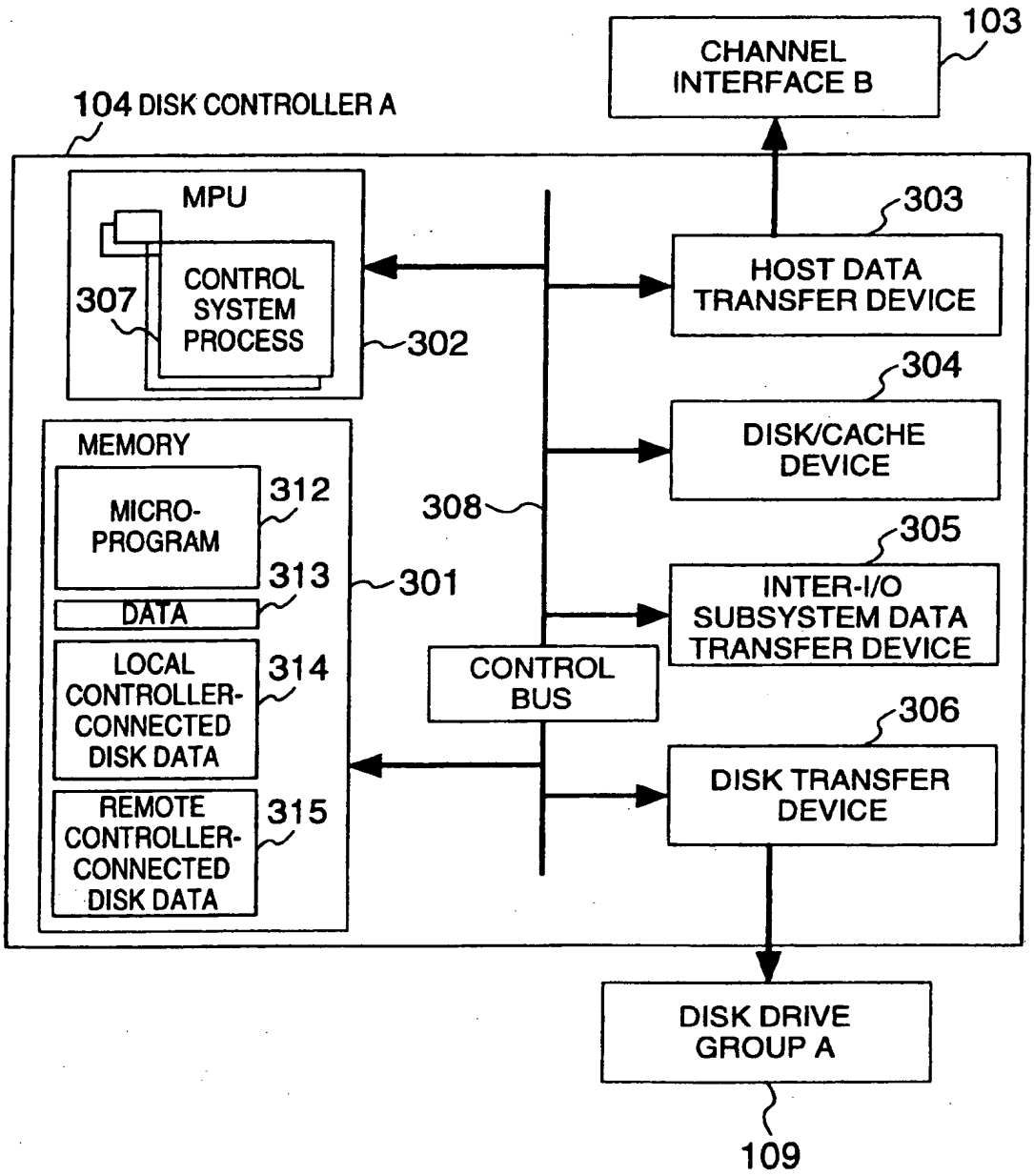

CHANNEL INTERFACE B
103
104 DISK CONTROLLER A
MPU
CONTROL SYSTEM PROCESS
307
302
303
HOST DATA TRANSFER DEVICE
304
DISK/CACHE DEVICE
MEMORY
MICRO-PROGRAM
312
313
DATA
314
LOCAL CONTROLLER-CONNECTED DISK DATA
315
REMOTE CONTROLLER-CONNECTED DISK DATA
308
301
305
INTER-I/O SUBSYSTEM DATA TRANSFER DEVICE
CONTROL BUS
306
DISK TRANSFER DEVICE
DISK DRIVE GROUP A
109

FIG.4

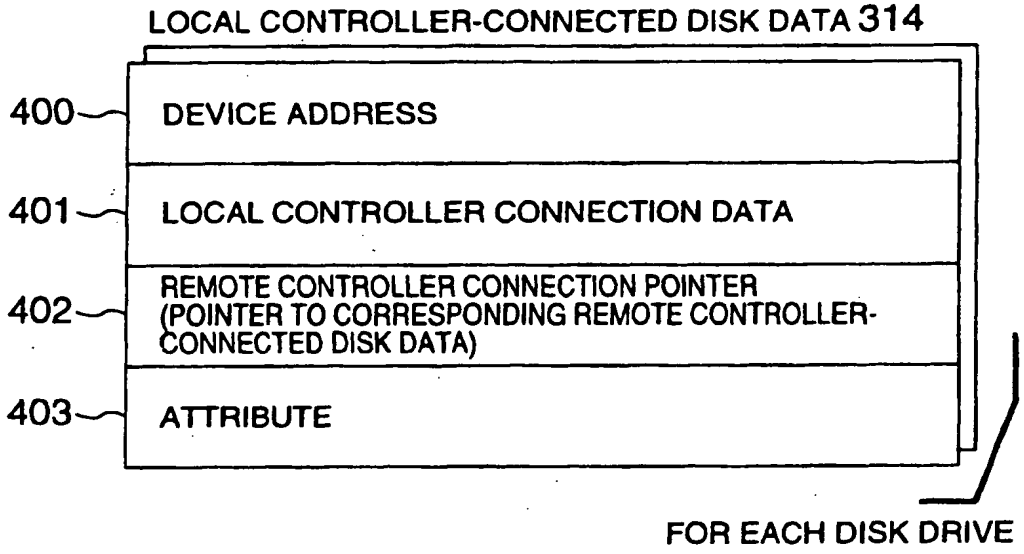

LOCAL CONTROLLER-CONNECTED DISK DATA 314
400
DEVICE ADDRESS
401
LOCAL CONTROLLER CONNECTION DATA
402
REMOTE CONTROLLER CONNECTION POINTER (POINTER TO CORRESPONDING REMOTE CONTROLLER-CONNECTED DISK DATA)
403
ATTRIBUTE
FOR EACH DISK DRIVE

FIG.5

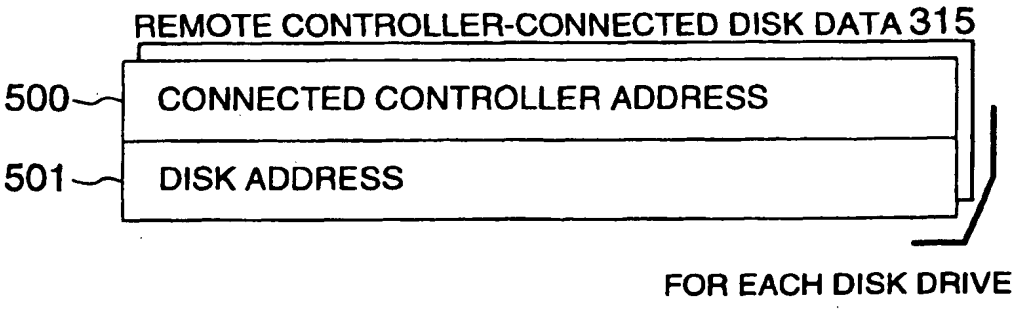

REMOTE CONTROLLER-CONNECTED DISK DATA 315
500
CONNECTED CONTROLLER ADDRESS
501
DISK ADDRESS
FOR EACH DISK DRIVE

FIG.6

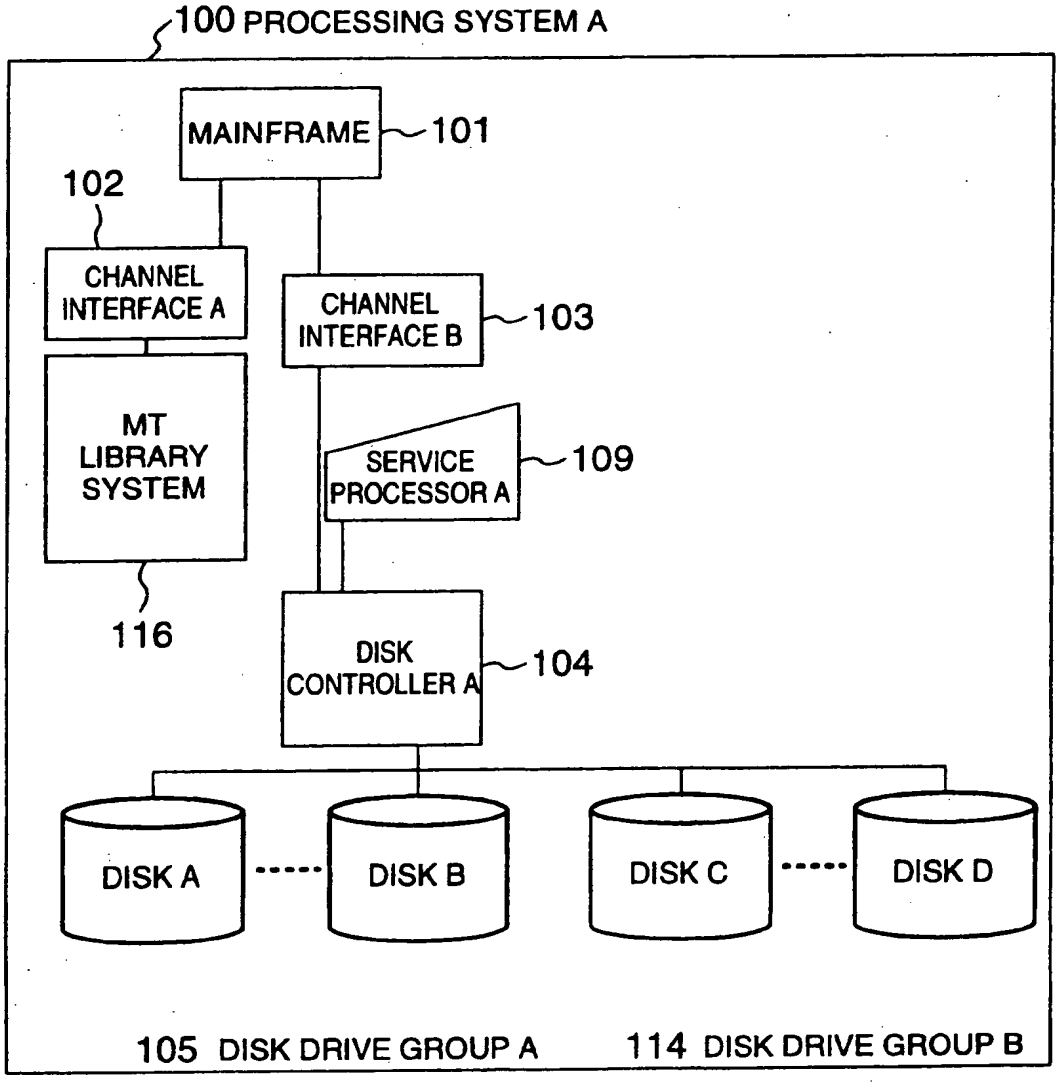

100 PROCESSING SYSTEM A
MAINFRAME
101
102
CHANNEL INTERFACE A
CHANNEL INTERFACE B
103
MT LIBRARY SYSTEM
SERVICE PROCESSOR A
109
116
DISK CONTROLLER A
104
DISK A
DISK B
DISK C
DISK D
105 DISK DRIVE GROUP A
114 DISK DRIVE GROUP B

FIG.7

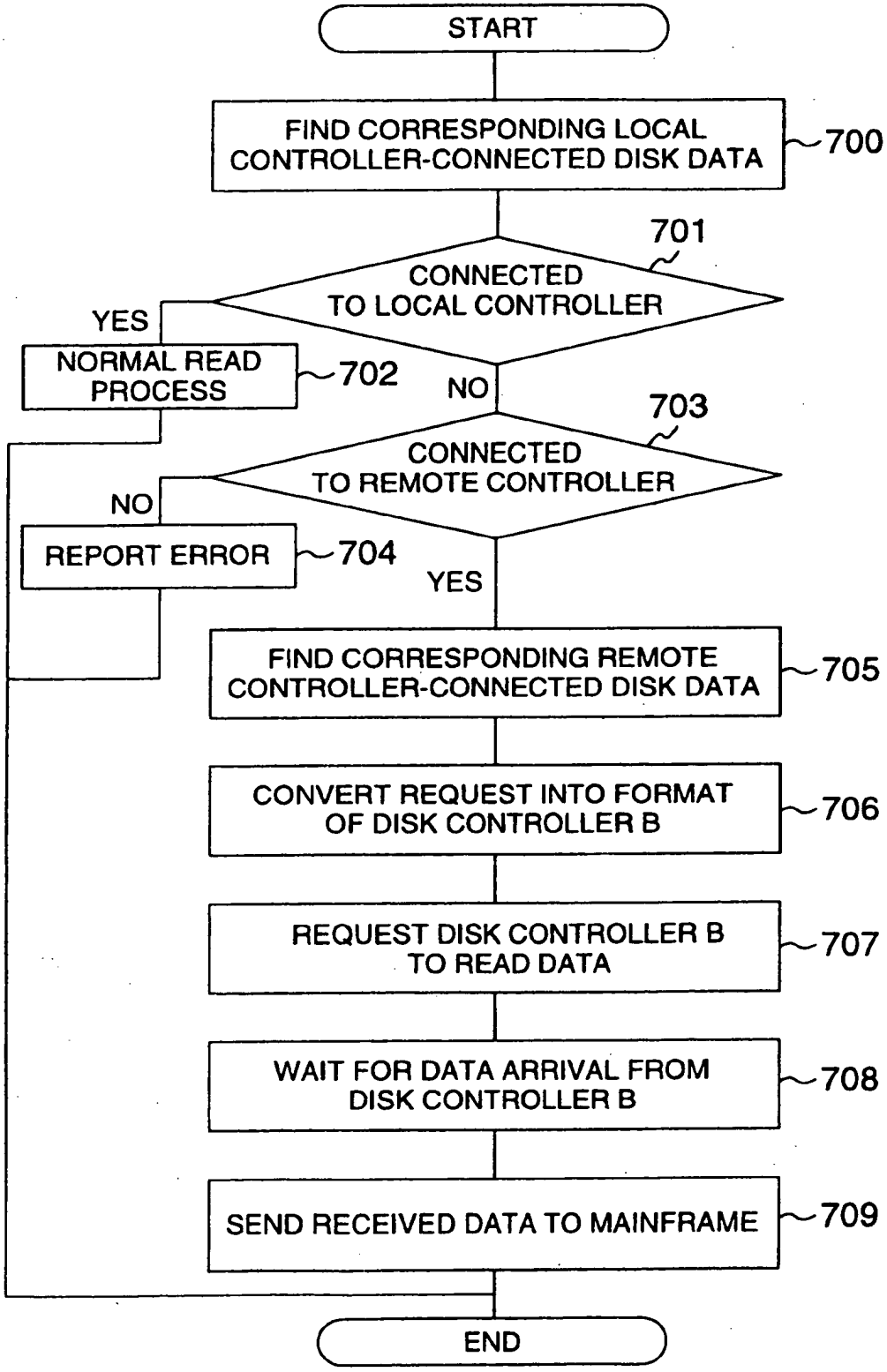
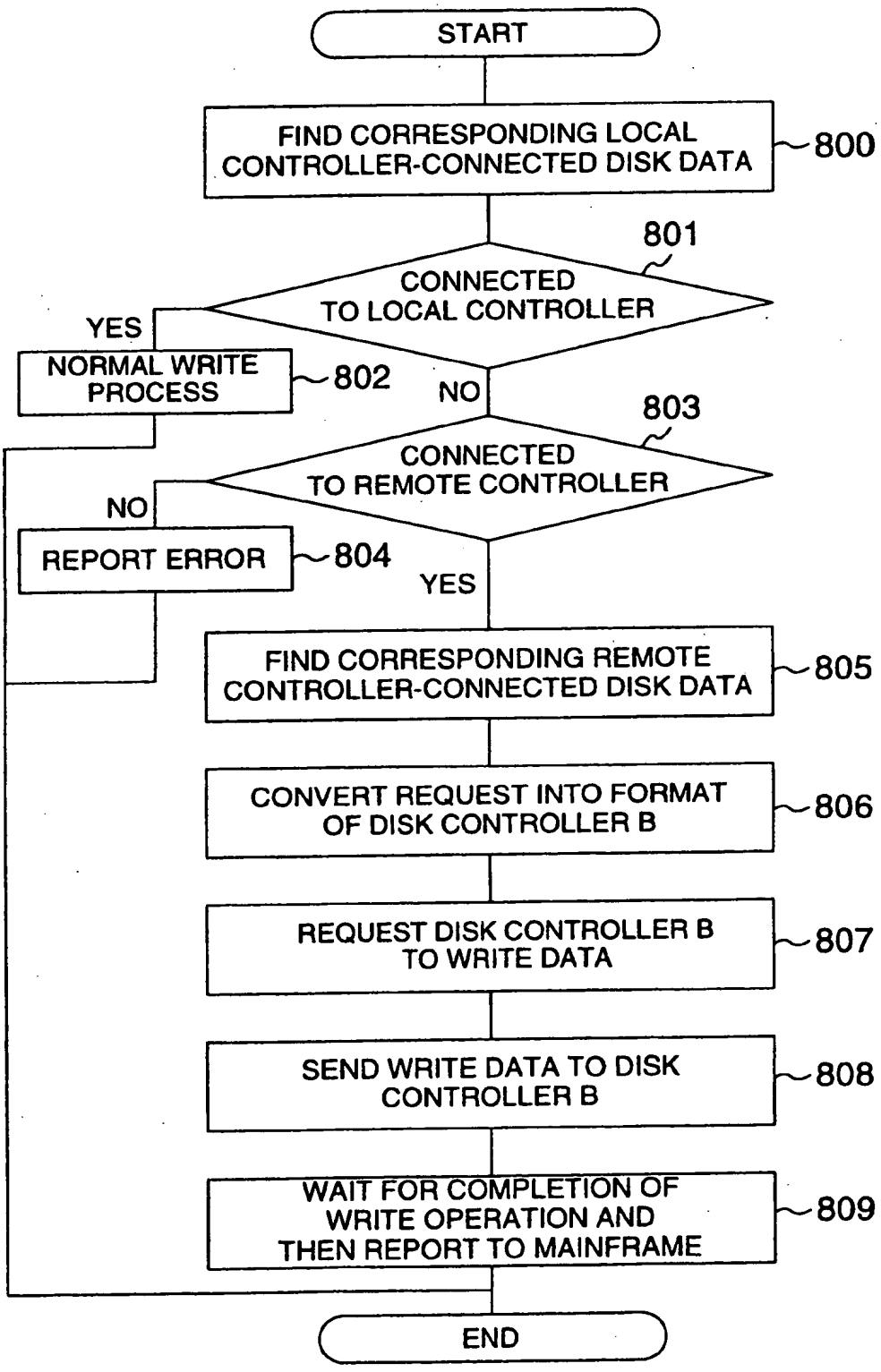

START
FIND CORRESPONDING LOCAL CONTROLLER-CONNECTED DISK DATA
700
701
CONNECTED TO LOCAL CONTROLLER
YES
NORMAL READ PROCESS
702
NO
703
CONNECTED TO REMOTE CONTROLLER
NO
REPORT ERROR
704
YES
FIND CORRESPONDING REMOTE CONTROLLER-CONNECTED DISK DATA
705
CONVERT REQUEST INTO FORMAT OF DISK CONTROLLER B
706
REQUEST DISK CONTROLLER B TO READ DATA
707
WAIT FOR DATA ARRIVAL FROM DISK CONTROLLER B
708
SEND RECEIVED DATA TO MAINFRAME
709
END

FIG.8


START
FIND CORRESPONDING LOCAL CONTROLLER-CONNECTED DISK DATA
800
801
CONNECTED TO LOCAL CONTROLLER
YES
NORMAL WRITE PROCESS
802
NO
803
CONNECTED TO REMOTE CONTROLLER
NO
REPORT ERROR
804
YES
FIND CORRESPONDING REMOTE CONTROLLER-CONNECTED DISK DATA
805
CONVERT REQUEST INTO FORMAT OF DISK CONTROLLER B
806
REQUEST DISK CONTROLLER B TO WRITE DATA
807
SEND WRITE DATA TO DISK CONTROLLER B
808
WAIT FOR COMPLETION OF WRITE OPERATION AND THEN REPORT TO MAINFRAME
809
END